CLAIM AMENDMENTS

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Currently Amended) A method, comprising:

loading a set of OS exception handler pointers into a first memory address space;

relocating the set of OS exception handler pointers to a second memory address space;

prior to OS runtime, loading a set of firmware-based exception filter pointers into the first address space to replace the set of OS exception handler pointers loaded into the first memory address space, wherein the firmware-based exception filter pointers point to exception filters that are stored in firmware;

vectoring an instruction pointer to a platform-specific firmware-based exception filter in response to an exception, wherein the platform-specific firmware-based exception filter is programmed in accordance with an architecture of a processor used to execute the platform-specific firmware-based exception filter.;

executing the firmware-based exception filter; and

re-vectoring the instruction pointer to an operating system (OS) exception handler configured to handle the exception.

2. (Original) The method of claim 1, wherein execution of the firmware-based exception filter performs operations including saving at least one processor register value to a storage device.

3. (Original) The method of claim 1, wherein execution of the firmware-based exception filter performs operations including saving at least a portion of system memory to a storage device.

4. (Currently Amended) The method of claim 1, ~~further comprising:~~

~~loading a set of OS exception handler pointers into a first memory address space;~~

~~relocating the set of OS exception handler pointers to a second memory address space; and~~

~~prior to OS runtime, loading a set of firmware-based exception filter pointers into the first address space~~ wherein the exception is thrown during OS runtime.

5. (Original) The method of claim 4, further comprising:

storing a base address of the second memory address space; and

employing the base address of the second memory address space to re-vector the instruction pointer to an OS exception handler pointer to the OS exception handler configured to handle the exception.

6. (Previously Presented) The method of claim 1, further comprising:

loading a set of OS exception handlers into a first memory address space;

relocating the set of OS exception handlers to a second memory address space; and

loading a set of firmware-based exception filters into the first address space, wherein the set of firmware-based exception filters comprise firmware-based exception handlers operates in a different execution regime than that of the operating system.

7. (Original) The method of claim 6, further comprising:

storing a base address of the second memory address space; and

employing the base address of the second memory address space to re-vector the instruction pointer to the OS exception handler configured to handle the exception.

8. (Previously Presented) The method of claim 1, further comprising:

loading a set of OS exception handler pointers into a first memory address space;

setting a processor exception vector register to include a base address of the first memory address space;

loading a set of firmware-based exception filter pointers into a second address space; and

replacing a base address of the first memory address space with the base address of the second memory address space in the processor exception vector register.

9. (Original) The method of claim 8, further comprising:

storing a base address of the first memory address space; and

employing the base address of the first memory address space to re-vector the instruction pointer to an OS exception handler pointer to the OS exception handler configured to handle the exception.

10. (Original) The method of claim 1, further comprising:

loading a set of OS exception handlers into a first memory address space;

setting a processor exception vector register to include a base address of the first memory address space;

loading a set of firmware-based exception filters into a second address space; and

resetting the processor exception vector register to include a base address of the second memory address space;

11. (Original) The method of claim 10, further comprising:

storing a base address of the first memory address space; and

employing the base address of the first memory address space to re-vector the instruction pointer to the OS exception handler configured to handle the exception.

12. (Previously Presented) The method of claim 1, further comprising:

loading the firmware-based exception filter into system memory; and

fixing up code in the firmware-based exception filter to re-vector the instruction pointer to one of the OS exception handler configured to handle the exception or a pointer to the OS exception handler configured to handle the exception.

13. (Previously Presented) A method, comprising:

loading a set of operating system (OS)-based exception handler components into system memory;

physically replacing within system memory entries for the set of OS-based exception handler components or logically replacing the set of OS-based exception handler components with a corresponding set of platform-specific firmware-based exception filter and/or handler components prior to OS runtime;

vectoring an instruction pointer to a firmware-based exception filter and/or handler in response to an OS runtime exception; and

executing the firmware-based exception filter and/or handler.

14. (Original) The method of claim 13, further comprising re-vectoring the instruction pointer to an operating system (OS) exception handler configured to handle the OS run-time exception after the firmware-based exception filter and/or handler has been executed.

15. (Original) The method of claim 14, further comprising fixing up code in the firmware-based exception filter and/or handler to re-vector the instruction pointer to one of the OS exception handler configured to handle the OS runtime exception or a pointer to the OS exception handler configured to handle the OS runtime exception.

16. (Original) The method of claim 13, wherein the set of OS-based exception handlers are physically replaced by:

copying the set of OS-based exception handlers from a physical address space to a virtual address space; and

overwriting the physical address space with the set of firmware-based exception filter and/or handler components.

17. (Original) The method of claim 13, wherein the set of OS-based exception handlers are logically replaced by:

loading the set of OS-based exception handlers into a first memory address space having a first base address; and

loading the set of firmware-based exception filter and/or handler components into a second address space having a second base address; and

replacing the first base address with the second base address in a register that is used to locate the base address of a table containing one of a set of exception handler procedures or pointers to a set of exception handler procedures.

18. (Previously Presented) A machine-readable recordable medium to provide instructions, which when executed perform operations including:

determining a first base address of a set of operating system (OS)-based exception handler components that have been loaded into a first memory address space;

storing the first base address;

loading a set of firmware-based exception filter and/or handler components into a second memory address space having a second base address; and

setting an exception vector register to have a base address corresponding to the second base address.

19. (Original) The machine-readable medium of claim 18, further to provide the set of firmware-based exception filter and/or handler components.

20. (Original) The machine-readable medium of claim 18, wherein the medium comprises a firmware storage device.

21. (Original) The machine-readable medium of claim 18, to provide further instructions to perform operations including:

filtering a runtime exception using a firmware-based exception filter; and

re-vectoring an instruction pointer to an operating system (OS) exception handler configured to handle the runtime exception.

22. (Previously Presented) A machine-readable recordable medium to provide instructions, which when executed perform operations including:

moving a set of operating system (OS)-based exception handler components from a first memory address space having a first base address to a second memory address space having a second base address;

storing the second base address; and

loading a set of firmware-based exception filter and/or handler components into the first memory address space.

23. (Original) The machine-readable medium of claim 22, further to provide the set of firmware-based exception filter and/or handler components.

24. (Original) The machine-readable medium of claim 22, wherein the medium comprises a firmware storage device.

25. (Original) The machine-readable medium of claim 22, to provide further instructions to perform operations including:

filtering a runtime exception using a firmware-based exception filter; and

re-vectoring an instruction pointer to an operating system exception handler configured to handle the runtime exception.

26. (Previously Presented) A system, comprising:

a processor;

memory, coupled to the processor;

a flash device, having firmware instructions stored thereon to perform operations in combination with logic programmed into the processor, the operations including:

loading a platform-specific firmware-based exception filter into memory;

detecting a runtime exception;

vectoring an instruction pointer to the firmware-based exception filter in response to the runtime exception;

executing the firmware-based exception filter; and

re-vectoring the instruction pointer to an operating system (OS) exception handler configured to handle the runtime exception.

27. (Original) The system of claim 26, further comprising a network interface coupled to the processor, wherein execution of firmware instructions loads a firmware-based exception filter from a network storage device via the network interface into the memory.

28. (Original) The system of claim 26, wherein execution of the firmware instructions performs further operations including:

determining a first base address of a set of OS-based exception handler components that have been loaded into a first address space of the memory;

storing the first base address;

loading a set of firmware-based exception filter and/or handler components into a second address space of the memory having a second base address; and

setting an exception vector register in the processor to have a base address corresponding to the second base address.

29. (Previously Presented) The system of claim 26, wherein execution of the firmware instructions perform the further operation of fixing up code in the firmware-based exception filter to re-vector the instruction pointer to one of the OS exception handler configured to handle the runtime exception or a pointer to the OS exception handler configured to handle the runtime exception.

30. (Original) The system of claim 26, wherein execution of the firmware instructions performs further operations including:

moving a set of OS-based exception handler components from a first address space in the memory having a first base address to a second address space in the memory having a second base address;

storing the second base address; and

loading a set of firmware-based exception filter and/or handler components into the first memory address space.